



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/259,361	02/26/1999	CLIFTON MALCOLM NOCK	RO998-203	5211
24038	7590	01/29/2004	EXAMINER	
MARTIN & ASSOCIATES, LLC			LAO, SUE X	
P O BOX 548			ART UNIT	
CARTHAGE, MO 64836-0548			PAPER NUMBER	

2126

10

DATE MAILED: 01/29/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary

Applicati n No.

09/259,361

Applicant(s)

NOCK ET AL.

Examiner

S. Lao

Art Unit

2126

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133).
- Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 10 November 2003.
- 2a) ☐ This action is **FINAL**. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 1-20 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☒ Claim(s) 15 is/are allowed.
- 6) ☒ Claim(s) 1-14, 16-20 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on _____ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.
- Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
- Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. §§ 119 and 120

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
 2. ☐ Certified copies of the priority documents have been received in Application No. _____.
 3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).
- * See the attached detailed Office action for a list of the certified copies not received.
- 13) ☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. § 119(e) (to a provisional application) since a specific reference was included in the first sentence of the specification or in an Application Data Sheet. 37 CFR 1.78.
- a) ☐ The translation of the foreign language provisional application has been received.
- 14) ☐ Acknowledgment is made of a claim for domestic priority under 35 U.S.C. §§ 120 and/or 121 since a specific reference was included in the first sentence of the specification or in an Application Data Sheet. 37 CFR 1.78.

Attachment(s)

- 1) ☒ Notice of References Cited (PTO-892)
- 2) ☒ Notice of Draftsperson's Patent Drawing Review (PTO-948)
- 3) ☐ Information Disclosure Statement(s) (PTO-1449) Paper No(s) _____.
- 4) ☐ Interview Summary (PTO-413) Paper No(s). _____.
- 5) ☐ Notice of Informal Patent Application (PTO-152)
- 6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1-20 are pending. This action is in response to applicant's appeal brief filed 11/10/2003.
2. The finality of the rejection of the last Office action is withdrawn upon further consideration of applicant's arguments filed 11/10/2003.
3. The text of those sections of Title 35, U.S. Code not included in this action can be found in a prior Office action.

4. Claims 1-7, 16-20 are rejected under 35 U.S.C. 103(a) as being unpatentable over De Borst et al (WO 98/02813) in view of Henckel et al (U S Pat. 5,864,866).

As to claim 16, Borst teaches (page 11, line 5 to page 14, line 31; page 19, line 12 to page 20, line 9 and page 22, lines 22-29) datastream factory that creates (Factory interface, TransportFactory interface), datastream class instance (Stream object, TransportStream object). Signal bearing media is inherent to the device of Borst.

Borst does not teach the data stream. Further, while Borst uses an identifier (object reference) to identify a stream object created by a Factory such that all subsequent stream operations must be addressed to this object (page 11, line 5 to page 14, line 31; page 19, line 12 to page 20, line 9 and page 22, lines 22-29), Borst does not teach that the identifier corresponds to a received data stream.

Henckel teaches a datastream receive mechanism (internalize() method) causing the data stream instance (person1 object) to populate itself (person1 object performs read_object() to read person2 object from the stream and then assigns it to a data member in person1 object, col. 13, lines 46-47 and col. 14, lines 24-26) with information (person2) contained in the datastream (stream) received from a second computer by invoking at least one object method on the instance (calls the internalize_from_stream () method on the person1 object). See col. 12, line 55 – col. 14, line 26. The datastream receive mechanism is defined in a datastream class in that the internalize_from_stream

Art Unit: 2126

() method is declared by the Streamable, i.e., the person class. Henckel further teaches the identifier of the datastream class instance (person1 object) corresponds to an identifier in a datastream received (original person1 objects of person class).

Therefore, it would have been obvious to allow the datastream factory of Borst to use the identifier corresponding to an identifier in a received datastream and allow the populating to use information contained in the received datastream in a self-populating manner. One of ordinary skill in the art would have been motivated to do so because this would have allowed objects with references to other objects be stored in streams and be referenced across processes (Henckel, col. 3, lines 18-24).

As to claim 17, Borst teaches recordable media (disk subsystem 43).

As to claim 18, Borst teaches transmission media (network connections 51, 53).

As to claim 19, Borst teaches datastream processing mechanism (Get, Put, Destroy and Cancel operations, page 12, lines 22-23).

As to claim 20, Borst teaches datastream send mechanism (Put operation).

As to claim 1, note the rejection of claim 16 above. Claim 1 additional recites processor, memory and second computer system. Borst teaches processor (CPU 45, 85), memory (memory 44, 84) and second computer system (client computer, server computer, gateways, pages 6, 7, 24). Borst also teaches Streaming can occur in both directions (page 27, line 27). The combined teaching of Borst and Henckel would have provided that the received datastream is one received from a second computer system.

As to claim 2, note the rejection of claim 19 above.

As to claim 3, note the rejection of claim 20 above.

As to claim 4, it is covered by claim 1 (datastream receive mechanism ... executed by).

As to claim 5, note the rejection of claim 19 above.

As to claim 6, note the rejection of claim 3 above.

As to claim 7, it is covered by claim 1 except for the first and second computers, network connection, and active datastream. Borst teaches first and second computers (multiple computers, pages 6-7) and network connection (path, page 19, lines 20-30; socket, page 22, lines 22-29). Note the received datastream of Borst as modified is an

Art Unit: 2126

active datastream because it contains updated object state data to be retrieved and processed at the receiver node.

5. Claims 8-14 are rejected under 35 U.S.C. 103(a) as being unpatentable over De Borst et al in view Henckel et al as applied to claims 1-7, 16-20 and further in view of Sosis ("Dynascope: A Tool for Program Directing").

As to claim 8, note the rejection of claim 1 for means' for creating and populating. Borst as modified by Henckel teaches means for constructing (Factory interface) and means for sending (Put operation, send on a socket, pages 19-20) a datastream.

Borst as modified does not teach that the datastream constructed and sent is an active datastream which identifies executable code for processing the active datastream.

Sosis teaches stream operations, wherein a datastream is an active datastream (executable stream) which identifies executable code for processing (program counter evpc, event instruction evinst) the active datastream (process event in the execution stream). See pages 13-14, section 2.2. Given the teaching of Sosis, it would have been obvious to include an active datastream into the datastreams being sent across the network of Borst. A motivation to do so would have been to provide extensibility and adaptability (Sosis, dealing with dynamics of program execution, page 12, section 1) desirable in Borst (Borst, page 4, lines 27-31).

As to claims 9 and 10, note the respective rejections of claims 2 and 3 above.

As to claim 11, it is covered by claim 8 except for method for communicating. Borst clearly teaches communicating method (delivering and receiving) between computers (abstract).

As to claim 12, note the rejection of claim 2 above.

As to claim 13, note the rejection of claim 16 (receive mechanism).

As to claim 14, note the rejection of claim 6 above.

6. Claims 1-7, 16-20 are rejected under 35 U.S.C. 103(a) as being unpatentable over De Borst et al (WO 98/02813) in view of Dyer et al (U S Pat. 5,754,849).

As to claim 16, Borst teaches (page 11, line 5 to page 14, line 31; page 19, line 12 to page 20, line 9 and page 22, lines 22-29) datastream factory that creates (Factory interface, TransportFactory interface), datastream class instance (Stream object, TransportStream object). Signal bearing media is inherent to the device of Borst.

Borst does not teach the data stream. Further, while Borst uses an identifier (object reference) to identify a stream object created by a Factory such that all subsequent stream operations must be addressed to this object (page 11, line 5 to page 14, line 31; page 19, line 12 to page 20, line 9 and page 22, lines 22-29), Borst does not teach that the identifier corresponds to a received data stream.

Dyer teaches causing a data stream instance (value object 104) to populate itself (reconstruct self) with information contained in the datastream received from a second computer (from flattened data stream 112) by invoking at least one object method on the instance (value.set method). See col. 12, lines 16-36; col. 35, lines 21-35; fig. 6, 3C. The datastream receive mechanism is defined in a datastream class (defined in value class, col. 3, lines 47-64) . Dyer further teaches the identifier of the datastream class instance corresponds to an identifier in a datastream received (class field 138 of value object, fig. 3C, col. 8, lines 22-28).

Therefore, it would have been obvious to allow the datastream factory of Borst to use the identifier corresponding to an identifier in a received datastream and allow the populating to use information contained in the received datastream in a self-populating manner. One of ordinary skill in the art would have been motivated to do so because this would have simplified transmission of a large variety of different types (col. 3, lines 16-37) of data from one node to another in the network of Borst.

As to claim 17, Borst teaches recordable media (disk subsystem 43).

As to claim 18, Borst teaches transmission media (network connections 51, 53).

As to claim 19, Borst teaches datastream processing mechanism (Get, Put, Destroy and Cancel operations, page 12, lines 22-23).

As to claim 20, Borst teaches datastream send mechanism (Put operation).

As to claim 1, note the rejection of claim 16 above. Claim 1 additionally recites processor, memory and second computer system. Borst teaches processor (CPU 45,

Art Unit: 2126

85), memory (memory 44, 84) and second computer system (client computer, server computer, gateways, pages 6, 7, 24). Borst also teaches Streaming can occur in both directions (page 27, line 27). The combined teaching of Borst and Dyer would have provided that the received datastream is one received from a second computer system.

As to claim 2, note the rejection of claim 19 above.

As to claim 3, note the rejection of claim 20 above.

As to claim 4, it is covered by claim 1 (datastream receive mechanism ... executed by).

As to claim 5, note the rejection of claim 19 above.

As to claim 6, note the rejection of claim 3 above.

As to claim 7, it is covered by claim 1 except for the first and second computers, network connection, and active datastream. Borst teaches first and second computers (multiple computers, pages 6-7) and network connection (path, page 19, lines 20-30; socket, page 22, lines 22-29). Note the received datastream of Borst as modified is an active datastream because it contains updated object state data to be retrieved and processed at the receiver node.

7. Claims 8-14 are rejected under 35 U.S.C. 103(a) as being unpatentable over De Borst et al in view of Dyer et al as applied to claims 1-7, 16-20 and further in view of Sosic ("Dynoscope: A Tool for Program Directing").

As to claim 8, note the rejection of claim 1 for means' for creating and populating. Borst as modified by Dyer teaches means for constructing (Factory interface) and means for sending (Put operation, send on a socket, pages 19-20) a datastream.

Borst as modified does not teach that the datastream constructed and sent is an active datastream which identifies executable code for processing the active datastream.

Sosic teaches stream operations, wherein a datastream is an active datastream (executable stream) which identifies executable code for processing (program counter evpc, event instruction evinst) the active datastream (process event in the execution stream). See pages 13-14, section 2.2. Given the teaching of Sosic, it would have been

Art Unit: 2126

obvious to include an active datastream into the datastreams being sent across the network of Borst. A motivation to do so would have been to provide extensibility and adaptability (Sobic, dealing with dynamics of program execution, page 12, section 1) desirable in Borst (Borst, page 4, lines 27-31).

As to claims 9 and 10, note the respective rejections of claims 2 and 3 above.

As to claim 11, it is covered by claim 8 except for method for communicating. Borst clearly teaches communicating method (delivering and receiving) between computers (abstract).

As to claim 12, note the rejection of claim 2 above.

As to claim 13, note the rejection of claim 16 (receive mechanism).

As to claim 14, note the rejection of claim 6 above.

8. Claims 1-7, 16-20 are rejected under 35 U.S.C. 103(a) as being unpatentable over De Borst et al (WO 98/02813) in view of Nakano et al ("Runtime Support for Extended Types in C++").

As to claim 16, Borst teaches (page 11, line 5 to page 14, line 31; page 19, line 12 to page 20, line 9 and page 22, lines 22-29) datastream factory that creates (Factory interface, TransportFactory interface), datastream class instance (Stream object, TransportStream object). Signal bearing media is inherent to the device of Borst.

Borst does not teach the data stream. Further, while Borst uses an identifier (object reference) to identify a stream object created by a Factory such that all subsequent stream operations must be addressed to this object (page 11, line 5 to page 14, line 31; page 19, line 12 to page 20, line 9 and page 22, lines 22-29), Borst does not teach that the identifier corresponds to a received data stream.

Nakano teaches causing a data stream instance (an instance corresponding to the type of an object read off the stream, such as TBizCard) to populate itself (stream-in operator resurrects name, picture and sound from the stream) with information contained in the datastream received from a second computer by invoking at least one object method on the instance (calls the object's stream-in operator to set the state of the object using the information read from the stream). See page 353-354, section 4.4.

Art Unit: 2126

In Nakano, the datastream receive mechanism is defined in a datastream class (TbizCard::operator<=(), page 354, left col., 2nd code listing) and the identifier of the datastream class instance corresponds to an identifier in a datastream received in that the type/identifier of the object is determined from the stream. Page 354, section 4.4, 1st para..

Therefore, it would have been obvious to allow the datastream factory of Borst to use the identifier corresponding to an identifier in a received datastream and allow the populating to use information contained in the received datastream in a self-populating manner. One of ordinary skill in the art would have been motivated to do so because this would have provided code portability (Nakano, page 355, right col., section 5) between different platforms/nodes of Borst.

As to claim 17, Borst teaches recordable media (disk subsystem 43).

As to claim 18, Borst teaches transmission media (network connections 51, 53).

As to claim 19, Borst teaches datastream processing mechanism (Get, Put, Destroy and Cancel operations, page 12, lines 22-23).

As to claim 20, Borst teaches datastream send mechanism (Put operation).

As to claim 1, note the rejection of claim 16 above. Claim 1 additionally recites processor, memory and second computer system. Borst teaches processor (CPU 45, 85), memory (memory 44, 84) and second computer system (client computer, server computer, gateways, pages 6, 7, 24). Borst also teaches Streaming can occur in both directions (page 27, line 27). The combined teaching of Borst and Nakano would have provided that the received datastream is one received from a second computer system.

As to claim 2, note the rejection of claim 19 above.

As to claim 3, note the rejection of claim 20 above.

As to claim 4, it is covered by claim 1 (datastream receive mechanism ... executed by).

As to claim 5, note the rejection of claim 19 above.

As to claim 6, note the rejection of claim 3 above.

As to claim 7, it is covered by claim 1 except for the first and second computers, network connection, and active datastream. Borst teaches first and second computers

Art Unit: 2126

(multiple computers, pages 6-7) and network connection (path, page 19, lines 20-30; socket, page 22, lines 22-29). Note the received datastream of Borst as modified is an active datastream because it contains updated object state data to be retrieved and processed at the receiver node.

9. Claims 8-14 are rejected under 35 U.S.C. 103(a) as being unpatentable over De Borst et al in view Nakano et al as applied to claims 1-7, 16-20 and further in view of Sosis ("Dynoscope: A Tool for Program Directing").

As to claim 8, note the rejection of claim 1 for means' for creating and populating. Borst as modified by Nakano teaches means for constructing (Factory interface) and means for sending (Put operation, send on a socket, pages 19-20) a datastream.

Borst as modified does not teach that the datastream constructed and sent is an active datastream which identifies executable code for processing the active datastream.

Sosis teaches stream operations, wherein a datastream is an active datastream (executable stream) which identifies executable code for processing (program counter evpc, event instruction evinst) the active datastream (process event in the execution stream). See pages 13-14, section 2.2. Given the teaching of Sosis, it would have been obvious to include an active datastream into the datastreams being sent across the network of Borst. A motivation to do so would have been to provide extensibility and adaptability (Sosis, dealing with dynamics of program execution, page 12, section 1) desirable in Borst (Borst, page 4, lines 27-31).

As to claims 9 and 10, note the respective rejections of claims 2 and 3 above.

As to claim 11, it is covered by claim 8 except for method for communicating. Borst clearly teaches communicating method (delivering and receiving) between computers (abstract).

As to claim 12, note the rejection of claim 2 above.

As to claim 13, note the rejection of claim 16 (receive mechanism).

As to claim 14, note the rejection of claim 6 above.

Art Unit: 2126

10. Claim 15 is allowed.

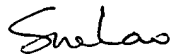
11. Applicant's arguments filed 11/10/2003 have been considered but are moot in view of the new ground(s) of rejection.

12. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

13. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Sue Lao whose telephone number is (703) 305-9657. A voice mail service is also available at this number. The fax phone number for the organization where this application or proceeding is assigned is (703) 872 9306.

Any inquiry of a general nature or relating to the status of this application or proceeding should be directed to the Group receptionist whose telephone number is (703) 305-9600.

Sue Lao



January 23, 2004